

# Web Engineering

## Lecture I

### **Web Fundamentals & Information Retrieval**

# Course ILOs

- Understand web search engines, evaluate and compare them.
- Modify search engines for specific applications.
- Provide broad coverage of the important issues in information retrieval and search engines.

# Course Outline

- Web Fundamentals
- Web Information Retrieval
- Architecture of Web Search Engines
- Tolerant Retrieval
- Index Construction
- Index Compression
- Scoring and term Weighting
- Vector Space Retrieval

# Course Assessments

■ Final Exam	50
■ Midterm Exam	15
■ Lab Quiz	5
■ Lab Assignment	10
■ Web app project	20

# Textbooks

- Croft, W. Bruce, Donald Metzler, and Trevor Strohman. "*Search engines: Information retrieval in practice*", 2015.
- Akshi Kumar, "**Web Technology: Theory and Practice**", 2018.

# What is Web Engineering?

*Web Engineering* is concerned with the establishment and use of systematic approaches (concepts, methods, tools and techniques) to the successful development, deployment, and maintenance of high-quality Web applications

# Web $\neq$ Internet

- **Internet:** a physical network connecting millions of computers using the same protocols for sharing/transmitting information (TCP/IP)
- **World Wide Web (WWW):** a collection of interlinked multimedia documents that are stored on the internet and accessed using a common protocol (Http)
- Key distinction: internet is hardware, Web is software

# Web Architecture

- The Web is a client-server system.
- Web browsers act as clients, making requests to web servers.
- Web servers respond to requests with requested information and/or computation generated by the client.
- Web applications are usually implemented with two-tier, three-tier, or multitier (N-tier) architectures.
- Each tier is a platform (client or server) with a unique responsibility.

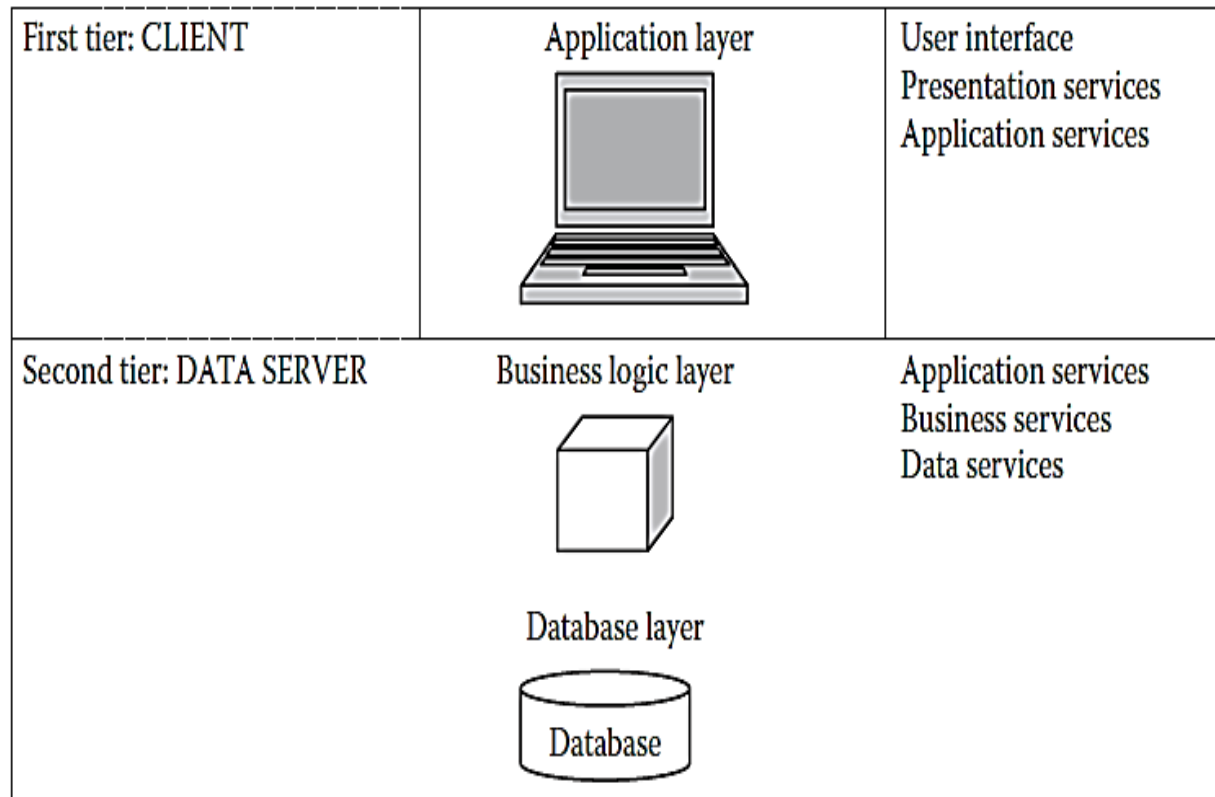


# Web Architecture

- In two-tier client-server architecture:



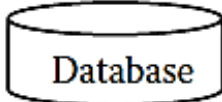
- Tier 1 is the **client** platform, hosting a web browser.

- Tier 2 is the **server** platform, hosting all the server software components



# Web Architecture

- In **three-tier** architecture, Tier 3 takes over part of the server function from Tier 2—typically data management.

First tier: CLIENT	Application layer 	User interface Presentation services
Second tier: APPLICATION SERVER	Business logic layer 	Application services Business services
Third tier: DATA SERVER	Data access layer  Database	Data services Data validation

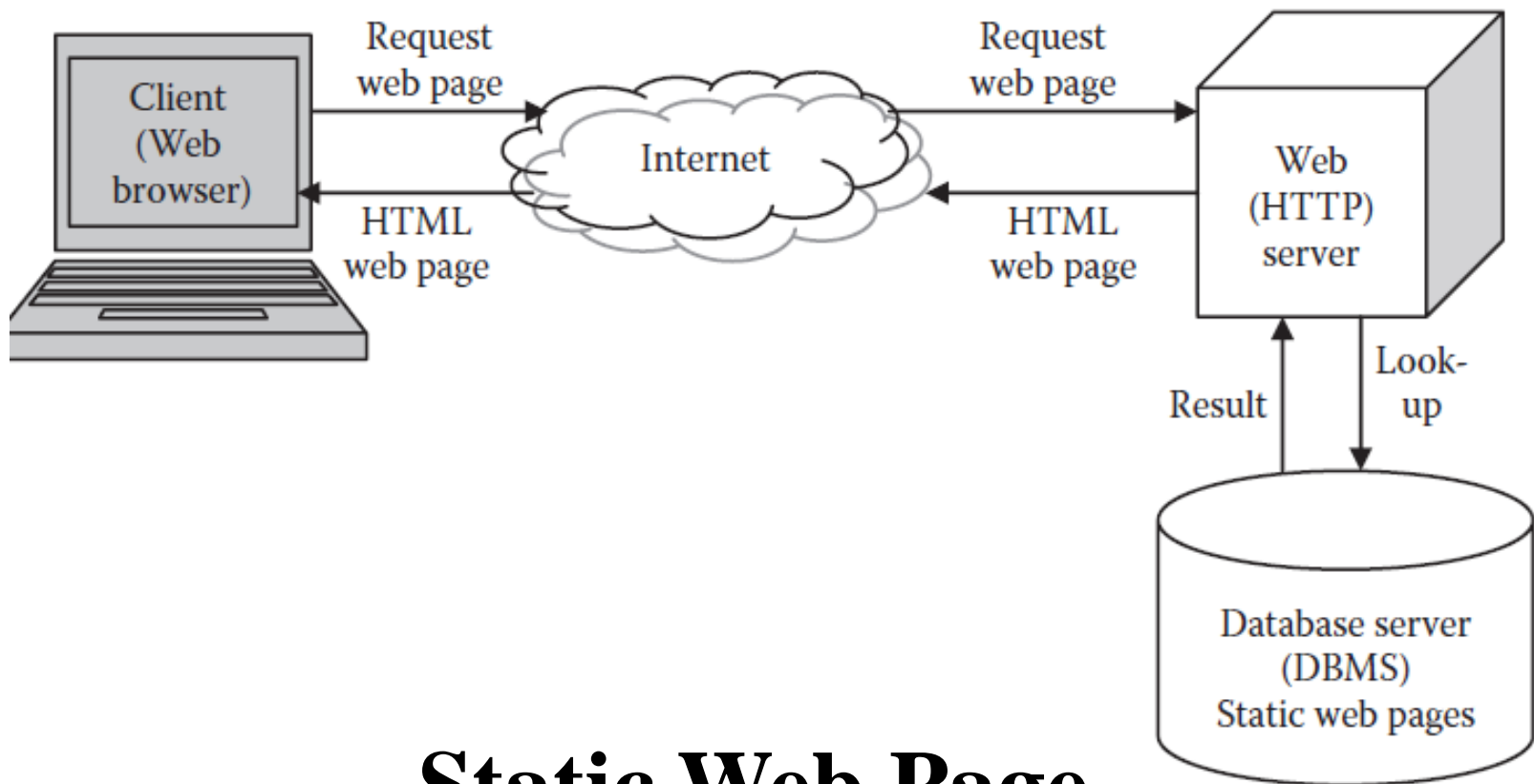
# Web Terminologies

- A web page is a document that can be displayed in a web browser.
- A website is a collection of web pages that are under one domain (news, scientific, entertainment, etc.)
- Web pages can be either *static* or *dynamic*.

# Web Terminologies

- **Static Web page** means the page is constant or unchanging.
  - Standard HTML pages are static web pages.
  - They contain HTML code, which defines the structure and content of the web page.
  - Each time an HTML page is loaded, it looks the same.
  - The only way the content of an HTML page will change is if a web developer edits and publishes an updated file.

# Web Terminologies

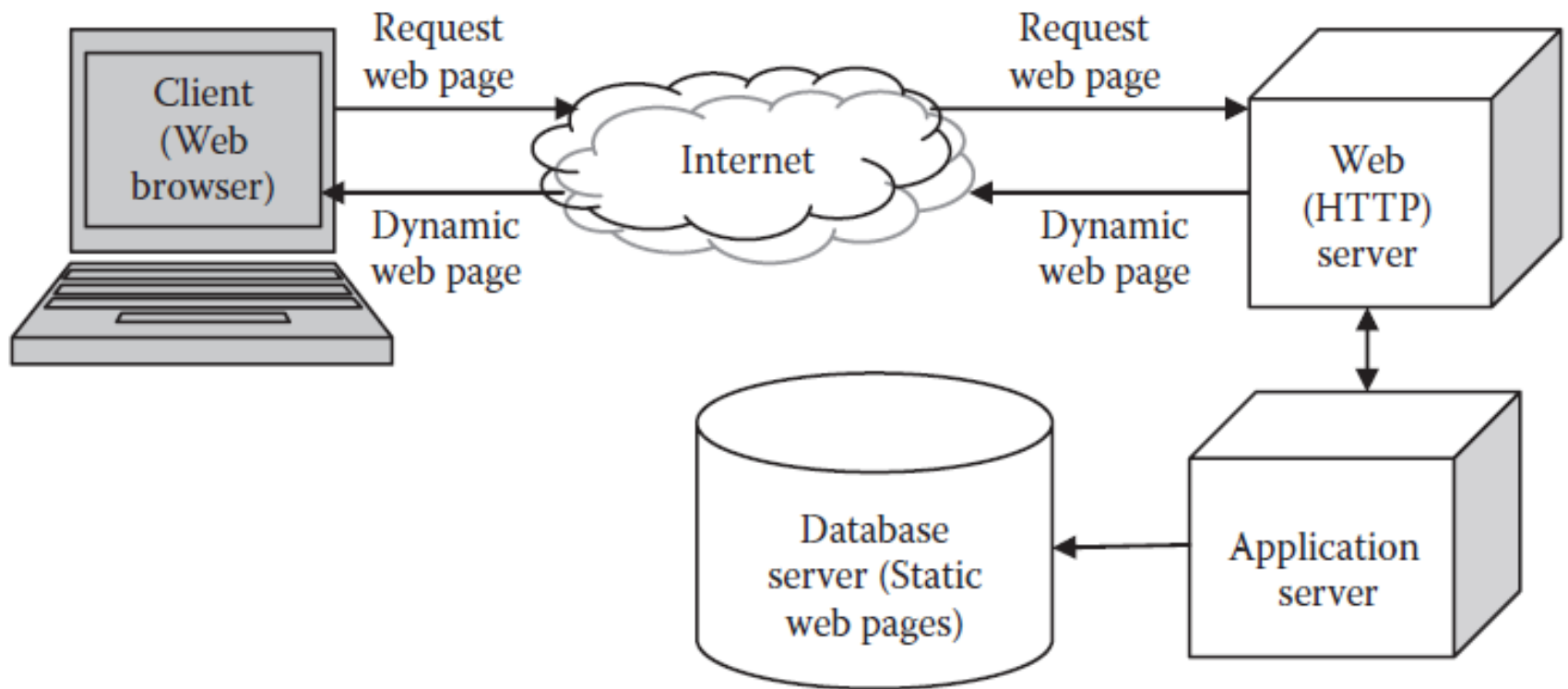


## Static Web Page

# Web Terminologies

- **Dynamic web pages**, on the other hand, contain server-side code such as PHP, ASP that allows the server to generate unique content each time the page is loaded.
  - For example, server might output a unique response based on a web form the user completed

# Web Terminologies



## Dynamic Web Page

# Websites Vs. Web Apps

- **Web sites** are primarily informational. For example, <http://cnn.com>.
- **Web applications** (or web apps), allow the user to perform actions.
  - Web apps are task-centric. For example, you might use your smartphone or tablet to find an app that accomplishes a specific task, like making a call, checking your email, or finding a taxi nearby.
  - Web apps are action-oriented rather than information-oriented.



# Web App Vs Web Service

- **Web services** provide a standard means of interoperating between different software applications running on a variety of platforms and frameworks.
  - A web site might use a web service. A company, for example, might provide a web page with a web application and a web service—a payment service like PayPal, for instance, has both a GUI for human users and a set of web services through which back-end systems can access the PayPal services.

Web app	Web service
an application that is accessed through a web browser running on client machine	system of software that allows different machines to interact with each other through a network
intended for Users (intended for human-to-computer interaction)	intended for other applications (intended for computer-to-computer interaction)
complete application with a user interface.	Does not necessarily have a UI since it is typically used as a component in an application

# Categories Of Web Applications

## (Based On Functionality)

<i>Functionality/Category</i>	<i>Examples</i>
Informational	Newspapers on net, manuals, reports, product catalogues, online books
Interactive	Registration forms, online games, customized information presentation
Transactional	Online shopping, online banking, online airline reservation, online bill payments
Workflow oriented	Online planning and scheduling, inventory management, status monitoring, software configuration management (SCM)

**WEB  
SEARCH  
ENGINES**

Information  
Retrieval

# Information Retrieval (IR)

- “*Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.*”
- Primary focus of IR has been on *text* and ***documents***.

# What Is A Document?

- Examples:
  - web pages, email, books, news stories, scholarly papers, text messages, Word, Powerpoint, PDF, forum postings, patents, IM sessions, etc.
- Common properties:
  - Significant text content.
  - Some meta data (e.g., title, author, date for papers; subject, sender, destination for email)

# Documents Vs. Database Records

- Database records (or *tuples* in relational databases) are typically made up of well-defined fields (or *attributes*)
  - e.g., bank records with account numbers, balances, names, addresses, social security numbers, dates of birth, etc.
- Easy to compare fields with well-defined semantics to queries in order to find matches.
- Text is more difficult.

# Documents Vs. Database Records

- Example bank database query:
  - *Find records with balance > \$50,000 in branches located in Amherst, MA.*
  - Matches easily found by comparison with field values of records
- Example search engine query:
  - *bank scandals in western mass*
  - This text must be compared to the text of entire news stories



# Comparing Text

- Comparing the query text to the document text and determining what is a good match is the core issue of information retrieval.
- Exact matching of words is not enough:
  - Many different ways to write the same thing in a “natural language” like English
  - e.g., does a news story containing the text “*bank director in Amherst steals funds*” match the query?
  - Some stories will be better matches than others

# Dimensions Of IR

- IR is more than just text, and more than just web search
  - Although these are central.
- People doing IR work with different media, different types of search applications, and different tasks.

Content	Applications	Tasks
Text	Web search	Ad hoc search
Images	Vertical search	Filtering
Video	Desktop search	Classification
Scanned docs	Forum search	Question answering
Audio	Literature search	
Music		

# IR Tasks

- Ad-hoc search
  - Find relevant documents for an arbitrary text query.
- Filtering
  - Identify relevant user profiles for a new document.
- Classification
  - Identify relevant labels for documents.
- Question answering
  - Give a specific answer to a question.

# Big Issues in IR

## ■ Relevance

- **Definition:** A relevant document contains the information that a person was looking for when they submitted a query to the search engine.
- Many factors influence a person's decision about what is relevant: e.g., task, context, novelty, style.
- *Topical relevance* (same topic) vs. *user relevance* (everything else).
- Retrieval models define a view of relevance.
- Ranking algorithms used in search engines are based on retrieval models.

# Big Issues In IR

## ■ Users and Information Needs

- Search evaluation is user-centered.
- Keyword queries are often poor descriptions of actual information needs.
- Interaction and context are important for understanding user intent.
- Query refinement techniques such as *query expansion*, *query suggestion*, *relevance feedback* improve ranking.

# Web Search & Information Retrieval

- Web search engines are practical application of information retrieval techniques to large scale text collections.
  - Search on the Web is a daily activity for many people throughout the world.
  - Search and communication are most popular uses of the computer.
  - Applications involving search are everywhere.
- Big issues in web search engines include main IR issues but also some others.

# IR And Search Engines

## Information Retrieval

### Relevance

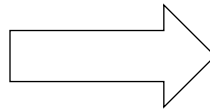
- Effective ranking*

### Evaluation

- Testing and measuring*

### Information needs

- User interaction*



## Search Engines

### Performance

- Efficient search and indexing*

### Incorporating new data

- Coverage and freshness*

### Scalability

- Growing with data and users*

### Adaptability

- Tuning for applications*

### Specific problems

- e.g. Spam*

# Search Engine Issues

## ■ Performance

- Measuring and improving the efficiency of search.
  - e.g., reducing *response time*, increasing *query throughput*, increasing *indexing speed*.
- *Indexes* are data structures designed to improve search efficiency.
  - designing and implementing them are major issues for search engines.



# Search Engine Issues

## ■ Dynamic data

- The “collection” for most real applications is constantly changing in terms of updates, additions, deletions
  - e.g., web pages
- Acquiring the documents is a major task
- Typical measures are *coverage* (how much has been indexed) and *freshness* (how recently was it indexed)
- Updating the indexes while processing queries is also a design issue

# Search Engine Issues

## ■ Scalability

- Making everything work with millions of users every day, and many terabytes of documents.
- Distributed processing is essential.

## ■ Adaptability

- Changing and tuning search engine components such as ranking algorithm, indexing strategy, interface for different applications.

# Search Engine Issues

## ■ SPAM

- For Web search, spam in all its forms is one of the major issues.
- Affects the efficiency of search engines and, more seriously, the effectiveness of the results.
- New subfield called *adversarial IR*, since spammers are “adversaries” with different goals.



**THANK YOU!**